NAG Library Routine Document

G01KFF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G01KFF returns the value of the probability density function (PDF) for the gamma distribution with shape argument α and scale argument β at a point x.

2 Specification

```
FUNCTION G01KFF (X, A, B, IFAIL)
REAL (KIND=nag_wp) G01KFF
INTEGER            IFAIL
REAL (KIND=nag_wp) X, A, B
```

3 Description

The gamma distribution has PDF

$$f(x) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)}x^{\alpha-1}e^{-x/\beta} \quad \text{if } x \geq 0; \quad \alpha, \beta > 0$$

$$f(x) = 0 \quad \text{otherwise.}$$

If $0.01 \leq x, \alpha, \beta \leq 100$ then an algorithm based directly on the gamma distribution's PDF is used. For values outside this range, the function is calculated via the Poisson distribution's PDF as described in Loader (2000) (see Section 9).

4 References

Loader C (2000) Fast and accurate computation of binomial probabilities **(not yet published)**

5 Arguments

1: X – REAL (KIND=nag_wp) *Input*

On entry: x, the value at which the PDF is to be evaluated.

2: A – REAL (KIND=nag_wp) *Input*

On entry: α, the shape argument of the gamma distribution.

Constraint: A > 0.0.

3: B – REAL (KIND=nag_wp) *Input*

On entry: β, the scale argument of the gamma distribution.

Constraints:

B > 0.0;
$\frac{\text{X}}{\text{B}} < \frac{1}{\text{X02AMF()}}$.

4: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

If IFAIL $\neq 0$, then G01KFF returns 0.0.

IFAIL $= 1$

On entry, A $= \langle value \rangle$.
Constraint: A > 0.0.

IFAIL $= 2$

On entry, B $= \langle value \rangle$.
Constraint: B > 0.0.

IFAIL $= 3$

Computation abandoned owing to overflow due to extreme parameter values.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

G01KFF is not threaded in any implementation.

9 Further Comments

Due to the lack of a stable link to Loader (2000) paper, we give a brief overview of the method, as applied to the Poisson distribution. The Poisson distribution has a continuous mass function given by,

$$p(x;\lambda) = \frac{\lambda^x}{x!}e^{-\lambda}. \tag{1}$$

The usual way of computing this quantity would be to take the logarithm and calculate,

$$\log(x;\lambda) = x\log\lambda - \log(x!) - \lambda.$$

For large x and λ, $x\log\lambda$ and $\log(x!)$ are very large, of the same order of magnitude and when calculated have rounding errors. The subtraction of these two terms can therefore result in a number, many orders of magnitude smaller and hence we lose accuracy due to subtraction errors. For example for $x = 2\times10^6$ and $\lambda = 2\times10^6$, $\log(x!) \approx 2.7\times10^7$ and $\log(p(x;\lambda)) = -8.17326744645834$. But calculated with the method shown later we have $\log(p(x;\lambda)) = -8.1732674441334492$. The difference between these two results suggests a loss of about 7 significant figures of precision.

Loader introduces an alternative way of expressing (1) based on the saddle point expansion,

$$\log(p(x;\lambda)) = \log(p(x;x)) - D(x;\lambda), \tag{2}$$

where $D(x;\lambda)$, the deviance for the Poisson distribution is given by,

$$\begin{aligned} D(x;\lambda) &= \log(p(x;x)) - \log(p(x;\lambda)), \\ &= \lambda D_0\left(\tfrac{x}{\lambda}\right), \end{aligned} \tag{3}$$

and

$$D_0(\epsilon) = \epsilon\log\epsilon + 1 - \epsilon.$$

For ϵ close to 1, $D_0(\epsilon)$ can be evaluated through the series expansion

$$\lambda D_0\left(\frac{x}{\lambda}\right) = \frac{(x-\lambda)^2}{x+\lambda} + 2x\sum_{j=1}^{\infty}\frac{v^{2j+1}}{2j+1}, \qquad \text{where } v = \frac{x-\lambda}{x+\lambda},$$

otherwise $D_0(\epsilon)$ can be evaluated directly. In addition, Loader suggests evaluating $\log(x!)$ using the Stirling–De Moivre series,

$$\log(x!) = \frac{1}{2}\log(2\pi x) + x\log(x) - x + \delta(x), \tag{4}$$

where the error $\delta(x)$ is given by

$$\delta(x) = \frac{1}{12x} - \frac{1}{360x^3} + \frac{1}{1260x^5} + \mathcal{O}\left(x^{-7}\right).$$

Finally $\log(p(x;\lambda))$ can be evaluated by combining equations (1)–(4) to get,

$$p(x;\lambda) = \frac{1}{\sqrt{2\pi x}}e^{-\delta(x)-\lambda D_0(x/\lambda)}.$$

10 Example

This example prints the value of the gamma distribution PDF at six different points X with differing A and B.

10.1 Program Text

```
    Program g01kffe

!     G01KFF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g01kff, nag_wp
```

```
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: a, b, f, x
      Integer                          :: ifail
!     .. Executable Statements ..
      Write (nout,*) 'G01KFF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Display titles
      Write (nout,*) '      X            A            B          RESULT'
      Write (nout,*)

d_lp: Do
        Read (nin,*,Iostat=ifail) x, a, b
        If (ifail/=0) Then
          Exit d_lp
        End If

        ifail = 0
        f = g01kff(x,a,b,ifail)

!       Display results
        Write (nout,99999) x, a, b, f
      End Do d_lp

99999 Format (1X,1P,4(1X,E12.4))
    End Program g01kffe
```

10.2 Program Data

```
G01KFF Example Program Data
1.0E-1 3.0E0 2.0E0
3.0E0 1.0E1 1.1E1
6.0E0 5.0E0 1.0E0
4.0E0 1.0E1 1.0E-1
9.0E0 9.0E0 5.0E-1
1.6E1 3.5E0 2.5E0                                          : X, A, B
```

10.3 Program Results

```
 G01KFF Example Program Results

       X            A            B          RESULT

    1.0000E-01   3.0000E+00   2.0000E+00   5.9452E-04
    3.0000E+00   1.0000E+01   1.1000E+01   1.5921E-12
    6.0000E+00   5.0000E+00   1.0000E+00   1.3385E-01
    4.0000E+00   1.0000E+01   1.0000E-01   3.0690E-08
    9.0000E+00   9.0000E+00   5.0000E-01   8.3251E-03
    1.6000E+01   3.5000E+00   2.5000E+00   2.0723E-02
```

Example Program
Plots of the Gamma Distribution